Electronic Version
Stylesheet Version v1.1.1

Description

METHOD FOR TRANSPARENT UPDATES OF OUTPUT DRIVER IMPEDANCE

BACKGROUND OF INVENTION

[0001] Field of the Invention

[0002] The present invention generally relates to solving the problem of updating output driver impedances in a way that is transparent to the user.

[0003] Description of the Related Art

[0004] High performance memories require reliable and well-behaved output driver impedances, and also require transparent impedance updates. As system temperature and voltages drift, periodic impedance updates occur to maintain a near-constant output driver impedance. However, the impedance updates must occur in a manner that does not disrupt bus operation. For example, a binarily-controlled output driver changing from an impedance value of "011111"to "100000"will see a discontinuity in

impedance as all bits change. Referring to the previous example, the target impedance of 50-Ohms may see an actual transition from 49-Ohms ("011111") to 120-Ohms ("000000") before returning to the ultimate value of 50-Ohms ("100000"). The impedance discontinuity is very undesirable and causes bus glitches and ultimately timing violations at the receiving device.

[0005] Another problem with conventional structures occurs in power-up cycles during initialization and periodicity of updates. Shifts in temperatures normally occur in periods that do not require very frequent updates. However, it is desired to have the optimum impedance after initialization cycles are completed. An impedance system employing Z binary bits require 2^Z impedance updates. If impedance updates occur every Y cycles, then the number of power-up cycles is $Y \times 2^Z$. If Z=7 and Y=256, then the number of power-up cycles becomes 32K, exceeding the application requirements of 1K.

[0006] The invention described below provides a structure and method that eliminates impedance discontinuity and improves startup performance.

SUMMARY OF INVENTION

[0007] The invention described below provides an output driver

for use with a static random access memory array. With the invention, a clock generator generates an output data path clock signal from a system clock signal. The output data path clock is timed differently than the system clock signal. A programmable impedance system is connected to the clock generator, and the programmable impedance system is timed according to the output data path clock signal. In addition a variable update circuit controls the programmable impedance system to perform impedance updates more frequently during initialization cycles than in cycles that occur after the initialization cycles expire.

[0008] The variable update circuit includes at least two differently timed clock dividers and a counter. There also are level translator circuits connected to the programmable impedance system. The level translator circuits each comprise a level translator and a pair of latches. The latches are connected to the clock generator and are timed according to the output data path clock signal. The invention also includes line driver circuits connected to different memory lines in the static random access memory array. Each of the line driver circuits comprises an output data latch, a pre-driver, a mid-driver, and a final stage driver; and the line driver circuits are connected to the

clock generator and are timed according to the output data path clock signal. Because the programmable impedance data path is triggered bythe clock generator and the line driver circuits are triggered by the clock generator, the timing of delivery of an impedance control signal from the programmable impedance data path circuit to the mid-driver is coordinated with the timing of delivery of data from the pre-driver to the mid-driver.

[0009] Thus, the invention controls the output driver by generating an output data path clock signal from a system clock signal and timing the programmable impedance of the output driver according to the output data path clock signal. The invention controls the timing of the line driver circuits according to the output data path clock signal. By timing the programmable impedance according to the output data path clock signal, the timing of delivery of an impedance control signal is coordinated with the timing of delivery of data. The invention also performs impedance updates on the output driver more frequently during initialization cycles than in cycles that occur after the initialization cycles expire using at least two differently timed clock dividers and a counter.

BRIEF DESCRIPTION OF DRAWINGS

[0010] The invention will be better understood from the following detailed description with reference to the drawings, in which: Figure 1 is a schematic diagram of a prior art driver circuit; Figure 2 is a schematic diagram of the present invention driver circuit; Figure 3 is a timing diagram showing the relationship between the impedance data and output driver pre-drive data; and Figure 4 is a flow diagram illustrating a preferred method of the invention.

DETAILED DESCRIPTION

[0011] Figure 1 shows one implementation of how impedance updates can occur. The high-level block diagram of the memory device in Figure 1 shows an output data latch 106 receiving data from the static random access memory (SRAM) array 104. The latch 106 is clocked in a single or double data rate with (DDR) generator 114 using rising and falling clock signals R_doclk and F_doclk. The true and complement data from the output data latch 106 (OCD_C/T) drives an output driver comprising a pre-drive and level translator circuit 108, a mid-driver 110, and a final stage 112. The final stage 112 comprises an array of binarily weighted P-type and N-type output driver devices. The output pin, DQ, is bi-directional.

[0012] Whenever a write occurs, signal HIZ from the controls and

address generator 102 places the output driver 108 in HIZ mode, which is the mode in which the driver impedance is updated. More specifically, the programmable impedance system 116 operates the L1 latch 120 to latch a new impedance value when in HIZ mode. The programmable impedance system 116 operates on a divided clock signal (divided by Y) produced by clock divider 118. The value latched in latch L1 is used by the level translator 122 to output Pdata and Ndata that, in turn, control the impedance of the mid-driver 110, and ultimately the impedance output by the driver. However, impedance updates can be issued only when the data bus 124 is not operational because the pre-driver 108 cannot simultaneously receive data and the HIZ signal. Further, timing variation between the HIZ signal and the clock make it very difficult to latch the impedance data (Pdata/Ndata).

[0013] The structure shown in Figure 1 has two substantial drawbacks. First, the outputs may never go to HIZ and the number of power-up cycles can be very large. The invention shown in Figure 2 provides a structure and method for allowing transparent impedance updates in separate I/O devices where the output driver may never see a HIZ condition. The invention also improves the power-up re-

quirements and periodicity of updates.

[0014] Figure 2 shows an inventive structure where transparent output impedance updates are achieved by timing the programmable impedance path similarly to the output data path. The clocking domain of the programmable impedance system 116 is changed from the main clock to the output latch clock R_doclk. Changing from the main clock to the output latch clock R_doclk solves two important problems. First, this structure removes any race conditions from the programmable impedance system to the L1 latch. The system drives global impedance data to several L1/L2 latches 120, 202. Each set of latches 210 services one of many groups of data lines 212. Impedance data for each L1/L2 latch 210 is locally captured by the identical R_doclk that launches the array data to the output driver. Therefore, the tracking between output data and programmable impedance data is well controlled.

[0015] With the invention shown in Figure 2, a clock divider 206 divides R_doclk to update the programmable impedance every 32 cycles (at power up), and every 256 cycles thereafter using two dividers 220 (256 divider) and 222 (32 divider), and a 10-bit counter. The logic gate 204 is controlled by the 10-bit counter to limit when the update to

the programmable impedance is made. The invention improves the power up requirements for optimum impedance. At power-up, the 10-bit counter 224 selects the path of a clock generated every 32 cycles 222. Thus, for the first 1024 initialization cycles, the programmable impedance is updated every 32 cycles, enough to reach near-optimum impedance. After the 1024 initialization cycles expire, the system is updated every 256 clock cycles by controlling logic device 204 to select the output from 256 clock divider 220. The slow-down in periodicity during normal use decreases the active current and unnecessary impedance updates.

[0016] As shown in Figure 2, the level translation circuitry 122 is placed before the L2 latch 202. This helps Pdata/Ndata signals arrive at the mid-driver 110 slightly before (e.g., less than ½clock cycle before) the data (N/P_DRV) arrives from the pre-drive 108. Thus, output driver impedance updates occur just prior to new data arriving. Therefore, impedance discontinuities at the time prior to driving data do not affect the signal integrity.

[0017] Figure 3 is a timing diagram showing the relationship between the impedance data (P/Ndata) and output driver Pre-drive data (P/N_DRV). Whenever output data clock

signal R_doclk pulses, both impedance data (Pdata/Ndata) and output data (OCD_C/OCD_T) are launched to the output driver. Signals Pdata/Ndata arrive at the Mid-Drive section of the Output Driver slightly ahead of data from the output data latch (P_DRV/N_DRV). The resulting waveforms provide an impedance update at a time when output data Q has been driven by the previous falling-edge clock transition (F_doclk). Thus the output data Q has been fully driven to a solid data state by almost a full half-cycle when the impedance update occurs. At the time of the impedance update, any discontinuity in output impedance does not affect the integrity of the data.

[0018] As shown in flowchart form in Figure 4, the invention controls the output driver by generating an output data path clock signal (40) from a system clock signal and times the programmable impedance of the output driver according to the output data path clock signal (42). Thus, the invention controls the timing of the line driver circuits according to the output data path clock signal (44). By timing the programmable impedance according to the output data path clock signal, the timing of delivery of an impedance control signal is coordinated with the timing of delivery of data (46). The invention also performs impedance updates

on the output driver more frequently during initialization cycles than in cycles that occur after the initialization cycles expire (48) using at least two differently timed clock dividers and a counter.

[0019] Thus, the invention described above provides an output driver for use with a static random access memory array. With the invention, a clock generator generates an output data path clock signal from a system clock signal. The output data path clock is timed differently than the system clock signal. A programmable impedance system is connected to the clock generator, and the programmable impedance system is timed according to the output data path clock signal. In addition a variable update circuit controls the programmable impedance system to perform impedance updates more frequently during initialization cycles than in cycles that occur after the initialization cycles expire.

[0020] The variable update circuit includes at least two differently timed clock dividers and a counter. There also are level translator circuits connected to the programmable impedance system. The level translator circuits each comprise a level transistor and a pair of latches. The latches are connected to the clock generator and are timed ac-

cording to the output data path clock signal. The invention also includes line driver circuits connected to different memory lines in the static random access memory array. Each of the line driver circuits comprises an output data latch, a pre-driver, a mid-driver, and a final stage driver; and the line driver circuits are connected to the clock generator and are timed according to the output data path clock signal. Because the level translation circuit is connected to the clock generator and the line driver circuits are connected to the clock generator, the timing of delivery of an impedance control signal from the level translation circuit to the mid-driver is coordinated with the timing of delivery of data from the pre-driver to the mid-driver.

[0021] Thus, the invention controls the output driver by generating an output data path clock signal from a system clock signal and timing the programmable impedance of the output driver according to the output data path clock signal. The invention controls the timing of the line driver circuits according to the output data path clock signal. By timing the programmable impedance according to the output data path clock signal, the timing of delivery of an impedance control signal is coordinated with the timing of

delivery of data. The invention also performs impedance updates on the output driver more frequently during initialization cycles than in cycles that occur after the initialization cycles expire using at least two differently timed clock dividers and a counter.

[0022] The invention provides transparent impedance updates to an output driver, which improves output data signal integrity. High-performance memory bus operations are limited by the signal characteristics of the input /output bus and driver and receiver circuits that operate the bus. This invention increases the bus performance by improving the signal integrity of the bus. Furthermore, the invention provides the optimum output driver impedance within the power-up initialization cycles provided to the memory.

[0023] While the invention has been described in terms of preferred embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.